



# UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE  
United States Patent and Trademark Office  
Address: COMMISSIONER FOR PATENTS  
P.O. Box 1450  
Alexandria, Virginia 22313-1450  
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/692,800	10/27/2003	Hideo Miyake	1450.1005D	1082

21171 7590 05/16/2007  
STAAS & HALSEY LLP  
SUITE 700  
1201 NEW YORK AVENUE, N.W.  
WASHINGTON, DC 20005

EXAMINER	
GEIB, BENJAMIN P	

ART UNIT	PAPER NUMBER
2181	

MAIL DATE	DELIVERY MODE
05/16/2007	PAPER

**Please find below and/or attached an Office communication concerning this application or proceeding.**

The time period for reply, if any, is set in the attached communication.

**Office Action Summary**

Application No.

10/692,800

Applicant(s)

MIYAKE ET AL.

Examiner

Benjamin P. Geib

Art Unit

2181

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

**Period for Reply**

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

**Status**

- 1) ☒ Responsive to communication(s) filed on 28 February 2007.
- 2a) ☐ This action is FINAL. 2b) ☒ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

**Disposition of Claims**

- 4) ☒ Claim(s) 14-34 is/are pending in the application.
- 4a) Of the above claim(s) \_\_\_\_\_ is/are withdrawn from consideration.
- 5) ☐ Claim(s) \_\_\_\_\_ is/are allowed.
- 6) ☒ Claim(s) 14-34 is/are rejected.
- 7) ☐ Claim(s) \_\_\_\_\_ is/are objected to.
- 8) ☐ Claim(s) \_\_\_\_\_ are subject to restriction and/or election requirement.

**Application Papers**

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☐ The drawing(s) filed on \_\_\_\_\_ is/are: a) ☐ accepted or b) ☐ objected to by the Examiner.  
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).  
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

**Priority under 35 U.S.C. § 119**

- 12) ☒ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☒ All b) ☐ Some \* c) ☐ None of:
1. ☒ Certified copies of the priority documents have been received.
  2. ☐ Certified copies of the priority documents have been received in Application No. \_\_\_\_\_.
  3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).
- \* See the attached detailed Office action for a list of the certified copies not received.

**Attachment(s)**

- |   |   |
|---|---|
| 1) <input checked="" type="checkbox"/> Notice of References Cited (PTO-892)   | 4) <input type="checkbox"/> Interview Summary (PTO-413)<br>Paper No(s)/Mail Date. _____ |
| 2) <input type="checkbox"/> Notice of Draftsperson's Patent Drawing Review (PTO-948)  | 5) <input type="checkbox"/> Notice of Informal Patent Application                       |
| 3) <input checked="" type="checkbox"/> Information Disclosure Statement(s) (PTO/SB/08)<br>Paper No(s)/Mail Date <u>02/28/2007</u> . | 6) <input type="checkbox"/> Other: _____  |

### DETAILED ACTION

1. Claims 14-34 have been examined.
2. It is hereby acknowledged that the following papers have been received and placed of record in the file: request for continued examination on 02/28/2007.
3. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 02/28/2007 has been entered.

### ***Claim Rejections - 35 USC § 112***

4. The following is a quotation of the second paragraph of 35 U.S.C. 112:  

The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.
5. Claims 14-33 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.
6. Regarding claims 14, 15, 21, 27, and 33, the preambles of the claims recite the limitation "a conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied and a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible" [emphasis added]. This limitation renders the claims indefinite as it unclear

Art Unit: 2181

how the designated data processing can be conditional upon "a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible" as indicated by the claim language (in particular, the words "when" and "and"). The above-cited limitation will be interpreted as "a conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied, wherein a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible" for the remainder of the examination as it appears to be what the applicant intended.

### ***Claim Rejections - 35 USC § 102***

7. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

A person shall be entitled to a patent unless –

(b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

(e) the invention was described in (1) an application for patent, published under section 122(b), by another filed in the United States before the invention by the applicant for patent or (2) a patent granted on an application for patent by another filed in the United States before the invention by the applicant for patent, except that an international application filed under the treaty defined in section 351(a) shall have the effects for purposes of this subsection of an application filed in the United States only if the international application designated the United States and was published under Article 21(2) of such treaty in the English language.

8. Claims 14-20, and 33 are rejected under 35 U.S.C. 102(b) as being anticipated by Krauskopf, U.S. Patent No. 5,165,027.

9. Referring to claim 14, Krauskopf has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional

instruction (*breakpoint instruction*) that executes a designated data processing when a designated condition of a branch is satisfied, wherein a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

a break detection section (*32-bit register and comparator; Fig. 2, component 34*) for detecting a breakpoint set at an arbitrary position of an instruction sequence [*The comparator detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register; See column 3, lines 43-63*];

a condition determination section (*enable logic circuit and control register; Fig. 2, components 36 and 32*) for determining whether a condition of a branch of said conditional instruction is satisfied [*The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of a branch of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39*]; and

a control section (*AND gate; Fig. 2, component 40*) for controlling a break-interrupt based upon of a breakpoint detection result from said break detection section and a branch condition determination result from said condition determination section [*The AND gate ANDs the hit signal and enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8*].

10. Referring to claim 15, Krauskopf has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction (*breakpoint instruction*) that executes a designated data processing when a designated condition of a branch is satisfied, wherein a determination of the condition of

Art Unit: 2181

the branch and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

an instruction break detection section (*32-bit register and comparator; Fig. 2, component 34*) for detecting an instruction break in accordance with whether an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is read, and outputting a detection signal (*hit signal; Fig. 2, component 46*) representing a detection result [*The comparator detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register; See column 3, lines 43-63*];

a condition determination section (*enable logic circuit and control register; Fig. 2, components 36 and 32*) for determining whether a condition of a branch of the conditional instruction is satisfied, and outputting a branch condition determination signal (*enable clock; Fig. 2, component 44*) [*The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of a branch of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39*]; and

a logical operation section (*AND gate; Fig. 2, component 40*) for performing AND operation to said detection signal output (*hit signal; Fig. 2, component 46*) from said instruction break detection section and said branch condition determination signal output (*enable clock; Fig. 2, component 44*) from said condition determination section, and sending a break-interrupt notification (*breakpoint/interrupt signal; See Fig. 2*) in accordance with the AND operation result [*The AND gate ANDs the hit signal and*

*enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8].*

11. Referring to claim 16, Krauskopf has taught an apparatus according to claim 15, wherein

said condition determination section (*enable logic circuit and control register*) is designed to determine whether an instruction word is said conditional instruction (*The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34*), if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal [*The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39*], and

when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, said logical operation section (*AND gate*) does not send the break-interrupt notification, and when said instruction word is the conditional instruction having a satisfied branch condition, said logical operation section sends said break-interrupt notification [*When the breakpoint has an unsatisfied condition the enable clock signal will not be set (column 4, lines 30-39) and, therefore, the AND gate will not send a break-interrupt notification. The Examiner notes that the claim language necessitates only that the logical operation section does not send a break-interrupt*

*notification when either the breakpoint is unconditional or conditional having an unsatisfied condition. When the breakpoint has a satisfied condition the enable clock signal will be set (column 4, lines 30-39). Since the hit and match signals will also be set, the AND gate will send a break-interrupt notification].*

12. Referring to claim 17, Krauskopf has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section (*control register; Fig. 2, component 32*) for setting one of a first mode (*breakpoint address represents a program reference*) in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the branch condition of said conditional instruction is satisfied, and a second mode (*breakpoint address represents a data reference*) in which said break-interrupt is generated when said generation condition of said instruction break is satisfied [*In both first and second modes a break-interrupt is generated when both the condition is satisfied (i.e. the clock enable is set) and the generation condition (i.e. the hit and match signals are set)*],

said condition determination section (*enable logic circuit and control register*) is designed to determine whether an instruction word is said conditional instruction (*The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34*), if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal [*The enable logic circuit determines whether the*

Art Unit: 2181

*breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39], and*

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, said logical operation section (*AND gate*) does not send the break-interrupt notification, and when said instruction word is the conditional instruction having a satisfied branch condition, said logical operation section sends the break-interrupt notification, and in said second mode, when said instruction word is an instruction word corresponding to the instruction address representing said breakpoint, said logical operation section (*AND gate*) sends said break-interrupt notification (*In both the first and second modes the operation of the logical operation section is the same; See arguments for last section of claim 16 regarding its operation*).

13. Referring to claim 18, Krauskopf has taught an apparatus according to claim 15, wherein

said condition determination section (*enable logic circuit and control register*) is designed to determine whether an instruction word is said conditional instruction (*The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34*), if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal [*The enable logic circuit determines whether the*

*breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39], and*

when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied branch condition, said logical operation section does not send the break-interrupt notification *[When the breakpoint has an unsatisfied condition the enable clock signal will not be set (column 4, lines 30-39) and, therefore, the AND gate will not send a break-interrupt notification]*, and when said instruction word is an unconditional instruction or the conditional instruction having a satisfied branch condition, said logical operation section sends said break-interrupt notification *[When the breakpoint has a satisfied condition the enable clock signal will be set (column 4, lines 30-39). Since the hit and match signals will also be set, the AND gate will send a break-interrupt notification. The Examiner notes that the claim language necessitates only that the logical operation section sends a break-interrupt notification when either the breakpoint is unconditional or conditional having a satisfied condition]*.

14. Referring to claim 19, Krauskopf has taught an apparatus according to claim 15, wherein

said apparatus further comprises a mode setting section *(control register; Fig. 2, component 32)* for setting one of a first mode *(breakpoint address represents a program reference)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the branch condition of said conditional instruction is satisfied, and a second mode *(breakpoint address represents a data*

*reference)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[In both first and second modes a break-interrupt is generated when both the condition is satisfied (i.e. the clock enable is set) and the generation condition (i.e. the hit and match signals are set)],*

said condition determination section (*enable logic circuit and control register*) is designed to determine whether an instruction word is said conditional instruction (*The enable logic circuit receives bus control signals which allow it to determine whether or not an instruction word is said conditional instruction (i.e. breakpoint); column 4, lines 30-34*), if said instruction word is said conditional instruction, determine whether the branch condition of the conditional instruction is satisfied, and output the branch condition determination signal *[The enable logic circuit determines whether the breakpoint is enabled (i.e. the condition is satisfied) using the information store in the control register; See column 4, lines 15-39]*, and

in said first mode, when an instruction word corresponding to the instruction address representing said breakpoint is a conditional instruction having an unsatisfied branch condition, said logical operation section does not send the break-interrupt notification, and when said instruction word is an unconditional instruction or the conditional instruction having a satisfied branch condition, said logical operation section sends the break-interrupt notification, and in said second mode, when said instruction word is an instruction word corresponding to the instruction address representing said breakpoint, said logical operation section sends said break-interrupt notification *(In both*

*the first and second modes the operation of the logical operation section is the same; See arguments for last section of claim 18 regarding its operation).*

15. Referring to claim 20, Krauskopf has taught an apparatus according to claim 15, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction (*See column 3, lines 9-24 and Fig. 1*), a long instruction word processor for parallel executing short instructions forming a long instruction word, and a parallel processor for parallel executing at least one basic instruction forming a variable-length instruction word.

16. Referring to claim 33, Krauskopf has taught an interrupt control method for controlling a break-interrupt in a data processing system having a function of executing a conditional instruction (*breakpoint instruction*) that executes a designated data processing when a designated condition of a branch is satisfied, wherein a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible, said method comprising the steps of:

detecting a breakpoint set at an arbitrary position of an instruction sequence [*The comparator (Fig. 2, component 34) detects a breakpoint by comparing the address (Fig. 2; component 19) with the addresses stored in the 32-bit debug register (Fig. 2, component 34); See column 3, lines 43-63*];

determining whether a condition of a branch of said conditional instruction is satisfied [*The enable logic circuit determines whether the breakpoint is enabled (i.e. a condition of a branch of the breakpoint is satisfied) using the information store in the control register; See column 4, lines 15-39*]; and

controlling the break-interrupt based upon the detecting of said breakpoint and the determining of the branch condition of said conditional instruction [*The AND gate ANDs the hit signal and enable clock signal to control the break-interrupt (i.e. breakpoint/interrupt signal; See Fig. 2) column 5, lines 1-8*].

17. Claims 21-32 are rejected under 35 U.S.C. 102(e) as being anticipated by Alverson et al., U.S. Patent No. 6,480,818 (Herein referred to as Alverson).

18. Referring to claim 21, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated condition of a branch is satisfied, wherein a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

an instruction break detection section (*target thread execution subroutine; Fig. 11, component 1100*) for detecting an instruction break in accordance with whether an instruction corresponding to an instruction address representing a breakpoint, which is set in a register, is read [*The break instruction, which corresponds to an address and is set in a register (See Fig. 4B), is inherently read out for instruction execution*], and sending a break-interrupt notification in accordance with the detecting of the instruction break [*The target thread execution subroutine detects an instruction break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler*

Art Unit: 2181

*subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11]; and*

*a control section (breakpoint handler subroutine; Fig. 12, component 1125) for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said instruction break detection section (target thread execution subroutine), determining whether a condition of a branch of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with the determining of the branch condition of the conditional instruction [The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66 ].*

19. Referring to claim 22, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction (*The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210*), and when said instruction word is said conditional instruction, determines whether the branch condition of said conditional instruction is satisfied (*See Fig. 12, component 1225*), and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler (*The breakpoint handler subroutine always eventually returns; See last step of Fig. 12*), and when said instruction word as said instruction break target is a conditional instruction having a

Art Unit: 2181

satisfied branch condition, performs said break-interrupt processing [*When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

20. Referring to claims 23 and 29, taking claim 23 as exemplary, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section (*nub thread execution routine; Fig. 5, component 500*) for setting one of a first mode (*mode when breakpoint set is a conditional breakpoint*) in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of the branch of said conditional instruction is satisfied, and a second mode (*mode when breakpoint set is an unconditional breakpoint*) in which said break-interrupt is generated when said generation condition of said instruction break is satisfied [*The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63], and*

in said first mode, said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction (*See Fig. 12, component 1210*), when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied (*See Fig. 12, component 1225*), when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler (*The*

Art Unit: 2181

*breakpoint handler subroutine always eventually returns; See last step of Fig. 12), and when said instruction word as said instruction break target is a conditional instruction having a satisfied branch condition, performs said break-interrupt processing [When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65], and*

*in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification [In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

21. Referring to claims 24 and 30, taking claim 24 as exemplary, Alverson has taught an apparatus according to claim 21, wherein said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction (*The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210*), when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied (*See Fig. 12, component 1225*), when said instruction word as said instruction break target is a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler (*The breakpoint handler subroutine always eventually returns; See last step of Fig. 12*), and when said instruction word as said instruction break target is an unconditional instruction or a

Art Unit: 2181

conditional instruction having a satisfied branch condition, performs said break-interrupt processing *[When the breakpoint is unconditional or conditional and the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65].*

22. Referring to claims 25 and 31, taking claim 25 as exemplary, Alverson has taught an apparatus according to claim 21, wherein

said apparatus further comprises a mode setting section *(nub thread execution routine; Fig. 5, component 500)* for setting one of a first mode *(mode when breakpoint set is a conditional breakpoint)* in which said break-interrupt is generated when a generation condition of said instruction break is satisfied, and the condition of the branch of said conditional instruction is satisfied, and a second mode *(mode when breakpoint set is an unconditional breakpoint)* in which said break-interrupt is generated when said generation condition of said instruction break is satisfied *[The nub thread execution routine sets a breakpoint mode by recording information indicating whether the inserted breakpoint is conditional or not; column 15, lines 43-63], and*

in said first mode, said control section determines, in said interrupt handler, whether an instruction word as an instruction break target is said conditional instruction *(See Fig. 12, component 1210)*, when said instruction word is said conditional instruction, determines whether the condition of the branch of said conditional instruction is satisfied *(See Fig. 12, component 1225)*, when said instruction word as said instruction break target is a conditional instruction having an unsatisfied branch

condition, returns from said interrupt handler (*The breakpoint handler subroutine always eventually returns; See last step of Fig. 12*), and when said instruction word as said instruction break target is an unconditional instruction or a conditional instruction having a satisfied branch condition, performs said break-interrupt processing [*When the condition is satisfied (i.e. true) or breakpoint is unconditional the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65*], and

in said second mode, said control section performs said break-interrupt processing when receiving said break-interrupt notification [*In the second mode the breakpoint set is unconditional and, therefore, the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65*].

23. Referring to claims 26 and 32, taking claim 26 as exemplary, Alverson has taught an apparatus according to claim 21, wherein said data processing system comprises one of a scalar processor for performing one unit of processing in accordance with one instruction (*See Fig. 3, component 101 and column 1, lines 20-40*), a long instruction word processor for parallel executing short instructions forming a long instruction word, and a parallel processor for parallel executing at least one basic instruction forming a variable-length instruction word.

24. Referring to claim 27, Alverson has taught an interrupt control apparatus applied to a data processing system having a function of executing a conditional instruction that executes a designated data processing when a designated condition of a branch is

Art Unit: 2181

satisfied, wherein a determination of the condition of the branch and the executed data processing when the branch condition is satisfied are indivisible, said apparatus comprising:

a software break detection section (*target thread execution subroutine; Fig. 11, component 1100*) for detecting a software break in accordance with whether a breakpoint instruction placed at an arbitrary position of an instruction sequence is executed (*See Fig. 4*), and sending a break-interrupt notification in accordance with the detection of the software break [*The target thread execution subroutine detects a software break (i.e. breakpoint; See Fig. 11, component 1110) and notifies the breakpoint handler subroutine (i.e. sends a break-interrupt notification); See column 21, lines 15-39 and Fig. 11*]; and

a control section (*breakpoint handler subroutine; Fig. 12, component 1125*) for, in an interrupt handler activated in accordance with said break-interrupt notification supplied from said software break detection section (*target thread execution subroutine*), determining whether a condition of a branch of said conditional instruction is satisfied, and controlling break-interrupt processing in accordance with the determining of the branch condition of the conditional instruction [*The breakpoint handler subroutine determines if a condition of the conditional instruction is satisfied (and, therefore, the breakpoint is valid) and notifies the nub (i.e. controls break-interrupt processing) if the condition is satisfied; See column 21, lines 51-66*].

25. Referring to claim 28, Alverson has taught an apparatus according to claim 27, wherein said control section determines, in said interrupt handler, whether an instruction

Art Unit: 2181

word as a software break target is said conditional instruction (*The breakpoint handler subroutine determines if whether or not the breakpoint instruction is conditional; See Fig. 12, component 1210*), and when said instruction word is said conditional instruction, determines whether the branch condition of said conditional instruction is satisfied (*See Fig. 12, component 1225*), and when said instruction word as said software break target is an unconditional instruction or a conditional instruction having an unsatisfied branch condition, returns from said interrupt handler (*The breakpoint handler subroutine always eventually returns; See last step of Fig. 12*), and when said instruction word as said software break target is a conditional instruction having a satisfied branch condition, performs said break-interrupt processing [*When the condition is satisfied (i.e. true) the breakpoint handler subroutine notifies the nub (See Fig. 12, component 1240), which performs break-interrupt processing; See column 22, lines 45-65*].

26. Claim 34 is rejected under 35 U.S.C. 102(b) as being anticipated by Alpert et al., U.S. Patent No. 5,740,413 (Hereinafter Alpert).

27. Referring to claim 34, Alpert has taught an apparatus comprising:

A controller [*debug circuitry*]

detecting a breakpoint set at an arbitrary position of an instruction sequence [*detecting a branch instruction when the branch breakpoint unit is enabled; column 6, lines 38-55*];

determining a branch of an instruction [*determining if the branch is taken; column 6, lines 38-55*]; and

controlling a break-interrupt based upon the detecting the breakpoint and the determining of the branch of the instruction [column 6, lines 38-55].

### ***Response to Arguments***

6. Applicants arguments filed on February 28, 2007, have been fully considered but they are not found persuasive.

7. Applicant argues the novelty/rejection of claims 14-33 on pages 11-14 of the remarks, in substance that:

"The amendments clarify that the claimed 'conditional instruction' differs from a breakpoint instruction by expressly providing that the claimed 'conditional instruction' is an instruction executing a designated data processing when a condition of a branch is satisfied. A breakpoint instruction does not include a determination of a condition of a branch. Accordingly, neither Kauskopf nor Alverson disclose executing the 'condition instruction' and controlling a break-interrupt as claimed." (last paragraph on page 12)

These arguments are not found persuasive for the following reasons:

Regarding the applicant's argument that a breakpoint instruction does not include a determination of a condition of a branch, the examiner respectfully disagrees. As understood by one of ordinary skill in the art, a branch transfers execution to a designated data processing. Since, as understood by one of ordinary skill in the art, a breakpoint transfers execution to a breakpoint handler, a breakpoint branches to a designated data processing (i.e. the breakpoint handler). Therefore, a conditional breakpoint includes a determination of a condition of a branch of the conditional breakpoint.

Since Kauskopf has taught a breakpoint that is conditional upon whether the breakpoint is enabled [Kauskopf; column 4, lines 15-39], Kauskopf has taught determining whether a condition of a branch of a conditional instruction (i.e. breakpoint instruction) is satisfied (i.e. whether the breakpoint is enabled).

Since Alverson has taught a breakpoint that is conditional upon a specified condition being true [Alverson; column 21, lines 51-66; Fig. 12], Alverson has taught determining whether a condition of a branch of a conditional instruction (i.e. breakpoint instruction) is satisfied (i.e. whether the specified condition of the breakpoint is true).

### ***Conclusion***

28. The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

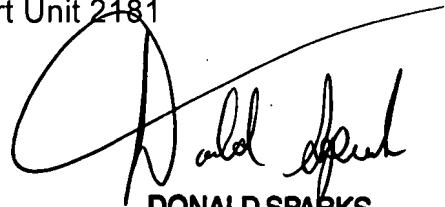
Any inquiry concerning this communication or earlier communications from the examiner should be directed to Benjamin P. Geib whose telephone number is (571) 272-8628. The examiner can normally be reached on Mon-Fri 8:30am-5:00pm.

Art Unit: 2181

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Donald Sparks can be reached on (571) 272-4201. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Benjamin P Geib  
Examiner  
Art Unit 2181

A handwritten signature in black ink, appearing to read "Donald Sparks", is written over a horizontal line.

**DONALD SPARKS**  
**SUPERVISORY PATENT EXAMINER**